

# Project Lead the Way Computer Science Essentials

Little Chute  
Math  
9th Grade, 1 Credit

## Course Overview/Description

Computer Science Essentials (CSE) is designed to be a full year (180-day) course implemented in the 9th grade. This course is an excellent entry point for new high school computer science (CS) learners. Students who have prior CS experiences will find ample opportunity to expand upon those experiences in this course. PLTW CSE introduces students to coding fundamentals through an approachable block-based programming language where they will have early success in creating usable apps. As students sharpen their computational thinking skills, they will transition to programming environments that reinforce coding fundamentals by displaying block programming and text based programming side by side. Finally, students will learn the power of text-based programming as they are introduced to the *Python* programming language.

## Course Essential Standards

### Scope and Sequence

Timeframe	Unit	Instructional Topics
45 Day(s)	<a href="#">Unit 1- Creative Computing: Building with Blocks</a>	1.1 Introduction to Computer Science Essentials 1.2 Collaborating around Computing 1.3 Innovation and Problem Solving
45 Day(s)	<a href="#">Unit 2- Computing and Society: Transitions to Text</a>	2.1 Transitions to Text-based Coding 2.2 Computing and Careers in our Society 2.3 Computing in our World
55 Day(s)	<a href="#">Unit 3- Solving with Syntax</a>	3.1 Collaborating in Text 3.2 Text-based Solutions 3.3 The Power of Text-based Programming
35 Day(s)	<a href="#">Unit 4- Computing with a Purpose</a>	4.1 Innovation of Computational Problem Solving

## UNIT 1: Creative Computing: Building with Blocks

**Duration of Unit: 45 Day(s)**

**Description of Unit:** Unit 1 welcomes new and returning students to the world of computer science and coding fundamentals. Students work with MIT App Inventor to create basic apps that rely on the concepts of event driven programming, branching, iteration, variables, and abstraction- the building blocks of creating with code. Students are introduced to essential computational thinking practices, such as developing abstractions, collaborating around computing, and communicating as they create, test, and refine computational artifacts of Android apps.

**Essential Questions and/or Enduring Understandings:**

1. Programming and the Design Process
2. Abstraction
3. Algorithms
4. Procedures
5. Variables
6. Lists
7. Loops, Conditionals, and Logic

ESSENTIAL Standards	Learning Targets
<b>1</b>	I can systematically design and iteratively develop a program for practical intent, personal expression, or to address a societal issue, while incorporating feedback from end users.
	I can decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
	I can systematically evaluate, test, and refine programs throughout the design process in order to make them more usable and accessible
	I can document programs in order to make them easier to follow, test, and debug.
	I can document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

<b>2</b>	I can describe the variety of abstractions used to represent data.
	I can abstraction to manage complexity in programs.
<b>3</b>	I can describe what an algorithm is and why algorithms are important within a program
	I can identify algorithms within a program and describe what function the algorithms serve
	I can construct my own algorithms within program of my creation
<b>4</b>	I can describe what a procedure is and why procedures are important in a program
	I can identify when a procedure is appropriate to use in a program that I am creating
	I can create a procedure in a program of my choice and appropriately send information to the procedure and access information from the procedure from within the program
<b>5</b>	I can name the different types of variables [boolean, string, integer, float] that can be used in a program
	I can determine the appropriate variables to be used in a program to accomplish a given purpose <ul style="list-style-type: none"> <li>• I can use a variable as an index to control how many times a program runs through a loop</li> <li>• I can use a variable to modify or access data in list</li> <li>• I can use a boolean variable to control the flow of a program</li> </ul>
	I can describe the difference between a global and a local variable and explain why and when is appropriate to use each in a program
<b>6</b>	I can identify a list in a program.
	I can determine when it would be appropriate to use a list in a program based on context
	I can create and modify a list in a program based on user input
	I can access items in a list from within a program to enhance the functionality of the program
<b>7</b>	I can describe the difference between “for” , “while” , and “repeat until” loops in terms of how they control the flow of a program
	I can choose the appropriate type of loop to apply within a program of my creation

	I can describe what it means to “nest” loops in a program
	I can create nested loops in a program of my own creation
	I can explain what a conditional is and describe the various types of conditionals [if-then, if-then-else]
	I can choose the appropriate type of conditional to apply within a program of my creation
	I can describe what it means to “nest” conditionals in a program
	I can create nested conditional in a program of my own creation
	I can use logical relationships [AND, OR, NOT] in combination with conditionals and loops to control the flow of a program.
<b>NICE TO KNOW Standards</b>	<b>Learning Targets</b>

## UNIT 2: Computing and Society: Transitions to Text

**Duration of Unit: 45 Day(s)**

**Description of Unit:** This unit continues to reinforce coding fundamentals as students are gradually introduced to text-based programming. In this unit, students will explore the impact of computer science on our society and bring coding off the screen and into the physical world. Students will learn how images can be used to make decisions in programs and explore real world applications and innovations that will shape our future.

**Essential Questions and/or Enduring Understandings:**

1. Programming and Problem Solving Process
2. Abstraction
3. Algorithms
4. Procedures
5. Variables
6. Lists
7. Loops, Conditionals, and Logic

ESSENTIAL Standards	Learning Targets
<b>1</b>	I can systematically design and iteratively develop a program for practical intent, personal expression, or to address a societal issue, while incorporating feedback from end users.
	I can decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
	I can systematically evaluate, test, and refine programs throughout the design process in order to make them more usable and accessible
	I can document programs in order to make them easier to follow, test, and debug.
	I can document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

<b>2</b>	I can describe the variety of abstractions used to represent data.
	I can abstraction to manage complexity in programs.
<b>3</b>	I can describe what an algorithm is and why algorithms are important within a program
	I can identify algorithms within a program and describe what function the algorithms serve
	I can construct my own algorithms within program of my creation
<b>4</b>	I can describe what a procedure is and why procedures are important in a program
	I can identify when a procedure is appropriate to use in a program that I am creating
	I can create a procedure in a program of my choice and appropriately send information to the procedure and access information from the procedure from within the program
<b>5</b>	I can name the different types of variables [boolean, string, integer, float] that can be used in a program
	I can determine the appropriate variables to be used in a program to accomplish a given purpose <ul style="list-style-type: none"> <li>• I can use a variable as an index to control how many times a program runs through a loop</li> <li>• I can use a variable to modify or access data in list</li> <li>• I can use a boolean variable to control the flow of a program</li> </ul>
	I can describe the difference between a global and a local variable and explain why and when is appropriate to use each in a program
<b>6</b>	I can identify a list in a program.
	I can determine when it would be appropriate to use a list in a program based on context
	I can create and modify a list in a program based on user input
	I can access items in a list from within a program to enhance the functionality of the program
<b>7</b>	I can describe the difference between “for” , “while” , and “repeat until” loops in terms of how they control the flow of a program
	I can choose the appropriate type of loop to apply within a program of my creation

	I can describe what it means to “nest” loops in a program
	I can create nested loops in a program of my own creation
	I can explain what a conditional is and describe the various types of conditionals [if-then, if-then-else]
	I can choose the appropriate type of conditional to apply within a program of my creation
	I can describe what it means to “nest” conditionals in a program
	I can create nested conditional in a program of my own creation
	I can use logical relationships [AND, OR, NOT] in combination with conditionals and loops to control the flow of a program.
<b>NICE TO KNOW Standards</b>	<b>Learning Targets</b>

## UNIT 3: Solving with Syntax

**Duration of Unit: 55 Day(s)**

**Description of Unit:** The goal of this unit is for students to begin to understand and use the flexibility and power of programming in a text-based environment. Students will be introduced to the *Python* programming language in the collaborative Replit development environment. In this unit, students will continue to build on coding fundamentals as they apply the same coding concepts, computational thinking practices, and development processes introduced in units 1 and 2.

**Essential Questions and/or Enduring Understandings:**

1. Programming and Problem Solving Process
2. Abstraction
3. Algorithms
4. Procedures
5. Variables
6. Lists
7. Loops, Conditionals, and Logic

ESSENTIAL Standards	Learning Targets
1	I can systematically design and iteratively develop a program for practical intent, personal expression, or to address a societal issue, while incorporating feedback from end users.
	I can decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
	I can systematically evaluate, test, and refine programs throughout the design process in order to make them more usable and accessible
	I can document programs in order to make them easier to follow, test, and debug.
	I can document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

<b>2</b>	I can describe the variety of abstractions used to represent data.
	I can abstraction to manage complexity in programs.
<b>3</b>	I can describe what an algorithm is and why algorithms are important within a program
	I can identify algorithms within a program and describe what function the algorithms serve
	I can construct my own algorithms within program of my creation
<b>4</b>	I can describe what a procedure is and why procedures are important in a program
	I can identify when a procedure is appropriate to use in a program that I am creating
	I can create a procedure in a program of my choice and appropriately send information to the procedure and access information from the procedure from within the program
<b>5</b>	I can name the different types of variables [boolean, string, integer, float] that can be used in a program
	<p>I can determine the appropriate variables to be used in a program to accomplish a given purpose</p> <ul style="list-style-type: none"> <li>• I can use a variable as an index to control how many times a program runs through a loop</li> <li>• I can use a variable to modify or access data in list</li> <li>• I can use a boolean variable to control the flow of a program</li> </ul>
	I can describe the difference between a global and a local variable and explain why and when is appropriate to use each in a program
<b>6</b>	I can identify a list in a program.
	I can determine when it would be appropriate to use a list in a program based on context
	I can create and modify a list in a program based on user input
	I can access items in a list from within a program to enhance the functionality of the program
<b>7</b>	I can describe the difference between “for” , “while”, and “repeat until” loops in terms of how they control the flow of a program

	I can choose the appropriate type of loop to apply within a program of my creation
	I can describe what it means to “nest” loops in a program
	I can create nested loops in a program of my own creation
	I can explain what a conditional is and describe the various types of conditionals [if-then, if-then-else]
	I can choose the appropriate type of conditional to apply within a program of my creation
	I can describe what it means to “nest” conditionals in a program
	I can create nested conditional in a program of my own creation
	I can use logical relationships [AND, OR, NOT] in combination with conditionals and loops to control the flow of a program.
<b>NICE TO KNOW Standards</b>	<b>Learning Targets</b>

## UNIT 4: Computing with a Purpose

**Duration of Unit: 35 Day(s)**

**Description of Unit:** The final unit of CSE allows students to apply all that they have learned in a student-defined, student-driven development. Students will apply computational thinking practices and a strategic development process to create computational artifacts that solve problems and create value for others. Students will collaborate the way computing professionals do as they pursue solutions to authentic needs. For those students continuing on to PLTW CSP, this unit provides an excellent model of how to participate in, document, and create a performance task for AP CSP.

**Essential Questions and/or Enduring Understandings:**

1. Programming and Problem Solving Process
2. Abstraction
3. Algorithms
4. Procedures
5. Variables
6. Lists
7. Loops, Conditionals, and Logic

ESSENTIAL Standards	Learning Targets
<b>1</b>	I can systematically design and iteratively develop a program for practical intent, personal expression, or to address a societal issue, while incorporating feedback from end users.
	I can decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
	I can systematically evaluate, test, and refine programs throughout the design process in order to make them more usable and accessible
	I can document programs in order to make them easier to follow, test, and debug.
	I can document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

<b>2</b>	I can describe the variety of abstractions used to represent data.
	I can abstraction to manage complexity in programs.
<b>3</b>	I can describe what an algorithm is and why algorithms are important within a program
	I can identify algorithms within a program and describe what function the algorithms serve
	I can construct my own algorithms within program of my creation
<b>4</b>	I can describe what a procedure is and why procedures are important in a program
	I can identify when a procedure is appropriate to use in a program that I am creating
	I can create a procedure in a program of my choice and appropriately send information to the procedure and access information from the procedure from within the program
<b>5</b>	I can name the different types of variables [boolean, string, integer, float] that can be used in a program
	I can determine the appropriate variables to be used in a program to accomplish a given purpose <ul style="list-style-type: none"> <li>• I can use a variable as an index to control how many times a program runs through a loop</li> <li>• I can use a variable to modify or access data in list</li> <li>• I can use a boolean variable to control the flow of a program</li> </ul>
	I can describe the difference between a global and a local variable and explain why and when is appropriate to use each in a program
<b>6</b>	I can identify a list in a program.
	I can determine when it would be appropriate to use a list in a program based on context
	I can create and modify a list in a program based on user input
	I can access items in a list from within a program to enhance the functionality of the program
<b>7</b>	I can describe the difference between “for” , “while”, and “repeat until” loops in terms of how they control the flow of a program

	I can choose the appropriate type of loop to apply within a program of my creation
	I can describe what it means to “nest” loops in a program
	I can create nested loops in a program of my own creation
	I can explain what a conditional is and describe the various types of conditionals [if-then, if-then-else]
	I can choose the appropriate type of conditional to apply within a program of my creation
	I can describe what it means to “nest” conditionals in a program
	I can create nested conditional in a program of my own creation
	I can use logical relationships [AND, OR, NOT] in combination with conditionals and loops to control the flow of a program.
<b>NICE TO KNOW Standards</b>	<b>Learning Targets</b>